Программирование

Система управления версиями git

Кафедра ИВТ и ПМ

2018

Система управления версий

Система управления версиями (Version Control System, VCS) — программное обеспечение для облегчения работы с изменяющейся информацией.
Позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Репозиторий, хранилище — место, где хранятся и поддерживаются какие-либо данные.

Начало работы

1. создать репозитарий

   `git init`

2. Добавить файлы к отслеживанию

   `git add файлы`

   Как правило в отслеживании должны быть файлы исходных кодов и другие файлы, необходимые для компиляции и запуска программы. Исполняемые файлы не отслеживаются. Потому, что их всегда можно получить после компиляции и чтобы не засорять ими репозитарий.

3. Просмотреть список отслеживаемых файлов.

   `git ls-files`

Типичный сценарий использования

Небольшие изменения.

- Внести изменения.
- Протестировать.

- Зафиксировать изменения (сделать коммит)

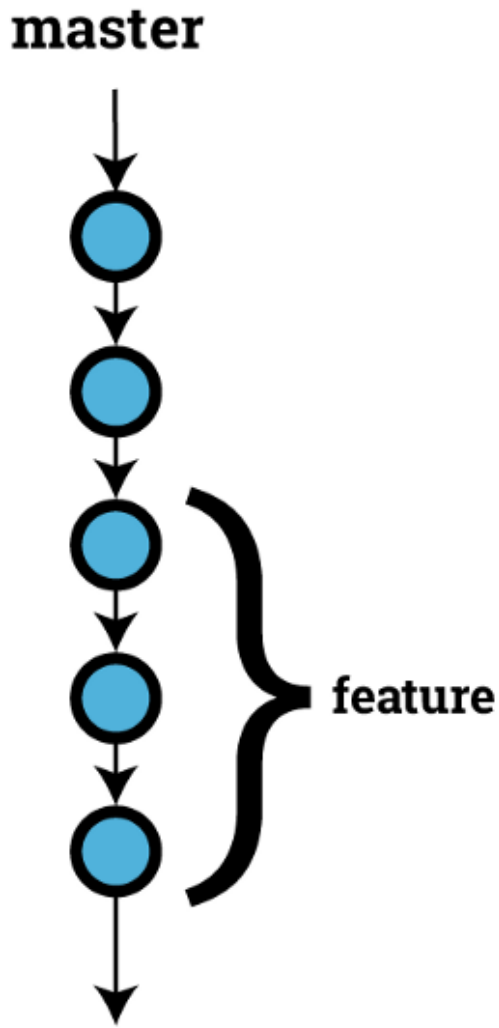
  `git commit -am "кратко об изменениях"`

  Ключи команды commit:
  -a - добавить все отслеживаемые файлы в фиксацию
  -m - комментарий к фиксации

  В комментариях следует кратко описывать сделанные изменения. Например: „добавлена функция foo()“ или „исправлен баг с отрисовкой героя“.

Типичный сценарий использования

Последовательность внесённых изменений



Внесение изменений

Фиксируемые изменения должны быть логически завершёнными.

Это означает, что после внесения изменений программа должна быть синтаксически правильной и работать корректно.

Нужно рассматривать комиты (внесение и фиксация изменений) как неделимые, атомарные действия в разработке программы.

Внесение изменений

Если планируются обширные изменения, то стоит подумать над созданием отдельной ветви, чтобы параллельно существовала исходная версия программы и версия, в которую вносятся изменения - *рабочая версия*.

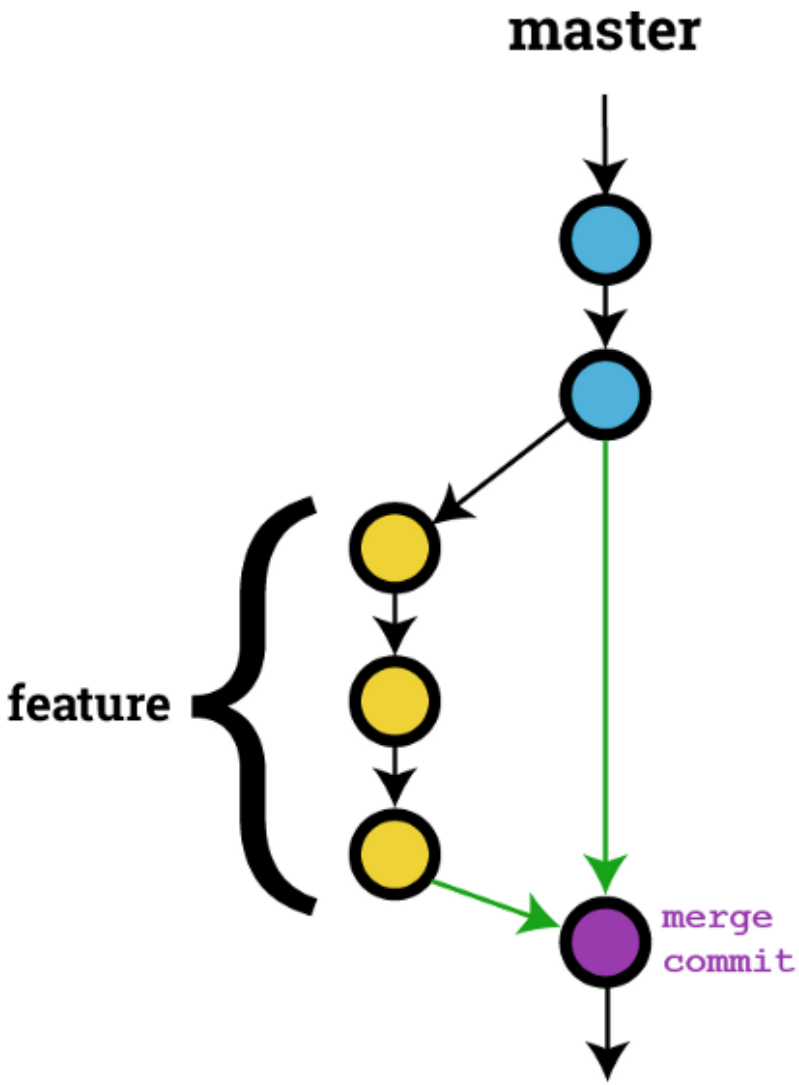
Допускается, что рабочая версия может не транслироваться, работать с ошибками.
Однако **в любой момент должна быть возможность вернутся к исправной версии программы**.

Внесение изменений

После того как изменения будет закончены, программа станет синтаксический правильной и протестирована, изменения рабочей версии добавляются к основной версии.

При внесении новых изменений, снова создаётся рабочая версия и всё повторяется заново.

Внесение изменений



Внесение изменений

В git отдельные версии программы хранятся в **ветках** (branches).

Одновременно с созданием репозитория создаётся основная ветка - **master**
Команды для работы с ветками **branch** <имя_ветки> -

команда создания ветки

checkout <имя_ветки> - переключения на ветку

checkout -b <имя_ветки> - создание ветки и переключение на неё

merge <имя>_

Переключится с ветки на ветку можно только если в текущей ветке все изменения зафиксированы.

Типичный сценарий использования

Значительные изменения

- Создать рабочую ветку и переключится

  `git checkout -b new_feature`

- Внести изменения
  - commit 1
  - commit 2
  - ....
  - commit n

- Переключится на основную ветку

  `git checkout master`

- Объединить основную ветку с рабочей

  `git merge new_feature`

Ссылки и литература

Ссылка на слайды

github.com/VetrovSV/Programming